08-31-00 A



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

August 30, 2000

Honorable Assistant
Commissioner of Patents
Washington, D.C. 20231

SUBJECT: Patent Application
Inventor: Paul S. Neuman
Title: Method for Improved First Level Cache Coherency
File No: RA 5290 (33012/289/101)

Dear Sir:

Enclosed herewith are the following papers comprising an application for patent as identified above:

1. Specification ( 17 pages)
2. Claims ( 6 pages)
3. Formal Drawings ( 4 pages)
4. Declaration and Power of Attorney
5. Assignment of Invention
6. Assignment Coversheet

Please charge the Assignment fee of $40.00 and the Patent Application filing fee of $768.00, calculated below, to Account No. 19-3790 of Unisys Corporation. If the calculated fee is incorrect, you are authorized to charge the correct fee.

The filing fee was calculated as follows:

| | | |
|---|---|---|
| 1. | Basic Fee | $690.00 |
| 2. | Additional Fees | |
| a. | Number of claims in excess of 20, ( 20-20=0 ) 0 times $18 | 0.00 |
| b. | Number of independent claims minus 3, ( 4-3=1 ) 1 times $78 | 78.00 |

TOTAL 768.00

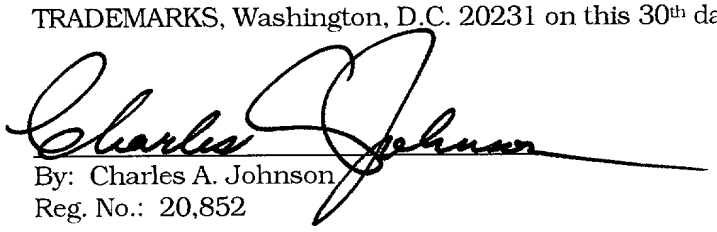
Correspondence is to be directed to the undersigned attorney of record, and an early acknowledgment will be greatly appreciated.

Respectfully submitted,

Charles A. Johnson
Reg. No. 20,852
Unisys Corporation
P O Box 64942 MS 4773
St. Paul, MN 55164
Attorney for Applicants

CAJ/rjn
(651)635-7702
Enclosures

CERTIFICATE UNDER 37 CFR 1.10: The undersigned hereby certifies that this transmittal letter and the paper of papers, as described hereinabove, are being deposited in the United States Postal Service, "Express Mail Post Office to Addressee" having an Express Mail mailing label number of EL 027383860 US, in an envelope addressed to: COMMISSIONER OF PATENTS AND TRADEMARKS, Washington, D.C. 20231 on this 30th day of August 2000.



By: Charles A. Johnson
Reg. No.: 20,852

**APPLICATION FOR UNITED STATES PATENT**

**INVENTOR:** Paul S. Neuman

**INVENTION:** METHOD FOR IMPROVED FIRST LEVEL CACHE COHERENCY

**DOCKET NUMBER:** RA 5290 (33012/289/101))

Unisys Corporation
Charles A. Johnson
P O Box 64942 - MS 4773
St. Paul, MN 55164
Attorney for Applicant
Reg. No.: 20,852

**SPECIFICATION**

CERTIFICATE UNDER 37 CFR 1.10: The undersigned hereby certifies that this transmittal letter and the paper of papers, as described hereinabove, are being deposited in the United States Postal Service, "Express Mail Post Office to Addressee" having an Express Mail mailing label number of EL 027383860 US, in an envelope addressed to: ASSISTANT COMMISSIONER OF PATENTS, Washington, D.C. 20231 on the 30th day of August, 2000.

Charles A. Johnson

# METHOD FOR IMPROVED FIRST LEVEL CACHE COHERENCY

## CROSS REFERENCE TO CO-PENDING APPLICATIONS

The present application is related to co-pending U.S. Patent Application Serial No. _____, filed _____, entitled Cooperative Hardware and Microcode Control System for Pipelined Instruction Execution; U.S. Patent Application Serial No. _____, filed _____, entitled Method for Avoiding Delays During SNOOP Requests; U.S. Patent Application Serial No. _____, filed _____, entitled Leaky Cache Mechanism; and U.S. Patent Application Serial No. _____, filed _____, entitled Data Coherency Protocol for Multi-level Cached High Performance Multiprocessor System, assigned to the assignee of the present invention and incorporated herein by reference.

## BACKGROUND OF THE INVENTION

1. **Field of the Invention:** - The present invention relates generally to data processing systems employing multiple instruction processors and more particularly relates to multiprocessor data processing systems employing multiple levels of cache memory.

2. **Description of the Prior Art:** - It is known in the art that the use of multiple instruction processors operating out of

common memory can produce problems associated with the processing of obsolete memory data by a first processor after that memory data has been updated by a second processor. The first attempts at solving this problem tended to use logic to lock processors out of memory spaces being updated. Though this is appropriate for rudimentary applications, as systems become more complex, the additional hardware and/or operating time required for the setting and releasing of locks can not be justified, except for security purposes. Furthermore, reliance on such locks directly prohibits certain types of applications such as parallel processing.

The use of hierarchical memory systems tends to further compound the problem of data obsolescence. U.S. Patent No. 4,056,844 issued to Izumi shows a rather early approach to a solution. The system of Izumi utilizes a buffer memory dedicated to each of the processors in the system. Each processor accesses a buffer address array to determine if a particular data element is present in its buffer memory. An additional bit is added to the buffer address array to indicate invalidity of the corresponding data stored in the buffer memory. A set invalidity bit indicates that the main storage has been altered at that location since loading of the buffer memory. The validity bits are set in accordance with the memory store cycle of each processor.

U.S. Patent No. 4,349,871 issued to Lary describes a bussed architecture having multiple processing elements, each having a dedicated cache memory. According to the Lary design, each

processing unit manages its own cache by monitoring the memory bus. Any invalidation of locally stored data is tagged to prevent use of obsolete data. The overhead associated with this approach is partially mitigated by the use of special purpose hardware and through interleaving the validity determination with memory accesses within the pipeline. Interleaving of invalidity determination is also employed in U.S. Patent No. 4,525,777 issued to Webster et al.

Similar bussed approaches are shown in U.S. Patent No. 4,843,542 issued to Dashiell et al, and in U.S. Patent No. 4,755,930 issued to Wilson, Jr. et al. In employing each of these techniques, the individual processor has primary responsibility for monitoring the memory bus to maintain currency of its own cache data. U.S. Patent No. 4,860,192 issued to Sachs et al, also employs a bussed architecture but partitions the local cache memory into instruction and operand modules.

U.S. Patent No. 5,025,365 issued to Mathur et al, provides a much enhanced architecture for the basic bussed approach. In Mathur et al, as with the other bussed systems, each processing element has a dedicated cache resource. Similarly, the cache resource is responsible for monitoring the system bus for any collateral memory accesses which would invalidate local data. Mathur et al, provide a special snooping protocol which improves system throughput by updating local directories at times not necessarily coincident with cache accesses. Coherency is assured

by the timing and protocol of the bus in conjunction with timing of the operation of the processing element.

An approach to the design of an integrated cache chip is shown in U.S. Patent No. 5,025,366 issued to Baror. This device provides the cache memory and the control circuitry in a single package. The technique lends itself primarily to bussed architectures. U.S. Patent No. 4,794,521 issued to Ziegler et al, shows a similar approach on a larger scale. The Ziegler et al, design permits an individual cache to interleave requests from multiple processors. This design resolves the data obsolescence issue by not dedicating cache memory to individual processors. Unfortunately, this provides a performance penalty in many applications because it tends to produce queuing of requests at a given cache module.

The use of a hierarchical memory system in a multiprocessor environment is also shown in U.S. Patent No. 4,442,487 issued to Fletcher et al. In this approach, each processor has dedicated and shared caches at both the L1 or level closest to the processor and at the L2 or intermediate level. Memory is managed by permitting more than one processor to operate upon a single data block only when that data block is placed in shared cache. Data blocks in dedicated or private cache are essentially locked out until placed within a shared memory element. System level memory management is accomplished by a storage control element through which all requests to shared main memory (i.e. L3 level) are routed. An apparent improvement to this approach is shown in U.S. Patent No.

4,807,110 issued to Pomerene et al. This improvement provides prefetching of data through the use of a shadow directory.

A further improvement to Fletcher et al, is seen in U.S. Patent No. 5,023,776 issued to Gregor. In this system, performance can be enhanced through the use of store around L1 caches used along with special write buffers at the L2 intermediate level. This approach appears to require substantial additional hardware and entails yet more functions for the system storage controller.

## SUMMARY OF THE INVENTION

The present invention overcomes the problems found in the prior art by providing a method and apparatus for improving the efficiency of maintaining coherency of a first level cache memory within a system having multiple levels of cache memory. This enhancement to efficiency is accomplished by utilizing a validation cycle, invalidation cycle, and parity detection circuitry, as discussed below.

The preferred mode of the present invention includes up to four main memory storage units. Each is coupled directly to each of up to four "pod"s. Each pod contains a level three cache memory coupled to each of the main memory storage units. Each pod may also accommodate up to two input/output modules.

Each pod may contain up to two sub-pods, wherein each sub-pod may contain up to two instruction processors. Each instruction processor has two separate level one cache memories (one for instructions and one for operands) coupled through a dedicated system controller, having a second level cache memory, to the level three cache memory of the pod.

Unlike many prior art systems, both level one and level two cache memories are dedicated to an instruction processor within the preferred mode of the present invention. The level one cache memories are of two types. Each instruction processor has an

instruction cache memory and an operand cache memory. The instruction cache memory is a read-only cache memory primarily having sequential access. The level one operand cache memory has read/write capability. In the read mode, it functions much as the level one instruction cache memory. In the write mode, it is a semi-store-in cache memory, because the level two cache memory is also dedicated to the instruction processor.

In accordance with the present invention, level one cache memory data is invalidated during the invalidate cycle under three conditions. In the third mode, a write hit causes invalidation as a result of declared ownership of the data. In the first or second modes, write hits or system bus snoop hits result in invalidation of the corresponding level one data.

When read data is made available, the system controller maintains a record of the location of the data within the level one cache memory.

Finally, a parity error within the system controller level two cache memory causes invalidation of the level one cache memory data. This is done as a precaution against loss of coherency control.

## BRIEF DESCRIPTION OF THE DRAWINGS

Other objects of the present invention and many of the attendant advantages of the present invention will be readily appreciated as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, in which like reference numerals designate like parts throughout the figures thereof and wherein:

**FIG. 1** is an overall block diagram of a fully populated system in accordance with the present invention;

**FIG. 2** is a schematic block diagram of one pod;

**FIG. 3** is a schematic block diagram of one instruction processor along with its dedicated system controller; and

**FIG. 4** is a detailed flow chart of the operation of the present invention.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

**FIG. 1** is an overall block diagram of fully populated data processing system according to the preferred mode of the present invention. This corresponds to the architecture of a commercial system of Unisys Corporation termed "Voyager".

The main memory of the system consists of up to four memory storage units, MSU 10, MSU 12, MSU 14, and MSU 16. Being fully modular, each of these four memory storage units is "stand-alone" and independent of one another. Each has a separate point-to-point dedicated bi-directional interface with up to four "pods", POD 18, POD 20, POD 22, POD 24. Again, each of the up to four pods is separate and independent of one another.

The contents of POD 20 are shown by way of example. For the fully populated system, POD 18, POD 22, and POD 24 are identical to POD 20. The interface between POD 20 and each of the four memory storage units (i.e., MSU 10, MSU 12, MSU 14, and MSU 16), is via a third level cache memory designated cached interface, CI 26, in this view. CI 26 couples with two input/output controllers, I/O Module 44 and I/O Module 46, and two sub-pods, SUB 28 and SUB 30. A more detailed explanation of the POD 20 is provided below.

The above described components are the major data handling elements of the system. In the fully populated system shown, there

are sufficient components of each type, such that no single hardware failure will render the complete system inoperative. The software employed within the preferred mode of the present system utilizes these multiple components to provide enhanced reliability for long term operation.

The remaining system components are utilitarian rather than data handling. System Oscillator 32 is the primary system time and clocking standard. Management System 34 controls system testing, maintenance, and configuration. Power Controller 36 provides the required electrical power. System Oscillator 38, Management System 40, and Power Controller 42 provide completely redundant backup capability.

**FIG. 2** is a more detailed block diagram of POD 20. The level three cache memory interfaces directly with the memory storage units via TLC Controller 26 (see also Fig. 1). The actual storage for the level three cache memory is TLC SRAMS 48. As indicated this static random access memory consists of eight 16 byte memory chips.

Subpod 28 and subpod 30 each contain up to two individual instruction processors. These are designated Voyager IP 50, Voyager IP 52, Voyager IP 54, and Voyager IP 56. As explained in detail below, each contains its own system controller. In accordance with the preferred mode of the present invention, these instruction processors need not all contain an identical software architecture.

**FIG. 3** is a more detailed block diagram of Voyager IP 50, located within Subpod 28, located within POD 20 (see also Figs. 1 and 2). As explained above, each instruction processor has a dedicated system controller having a dedicated level two cache memory. Instruction processor 64 has two dedicated level one cache memories (not shown in this view). One level one cache memory is a read-only memory for program instruction storage. Instruction processor 64 executes its instructions from this level one cache memory. The other level one cache memory (also not shown in this view) is a read/write memory for operand storage.

Instruction processor 64 is coupled via its two level one cache memories and dedicated system controller 58 to the remainder of the system. System controller 58 contains input logic 74 to interface with instruction processor 64. In addition, data path logic 70 controls movement of the data through system controller 58. The utilitarian functions are provided by Locks, Dayclocks, and UPI 62.

The remaining elements of system controller 58 provide the level two cache memory functions. SLC data ram 66 is the data actual storage facility. Control logic 70 provides the cache management function. SLC tags 72 are the tags associated with the level two cache memory. FLC-IC Dup. Tags 76 provides the duplicate tags for the level one instruction cache memory of instruction

processor 64. Similarly, FLC-OC Dup. Tags 78 provides the duplicate tags for the level one operand cache memory of instruction processor 64. For a more complete discusses of this duplicate tag approach, reference may be made with the above identified co-pending and incorporated U.S. Patent Applications.

**FIG. 4** is a detailed flow chart of the operation of the present invention showing three specific opportunities to improve efficiency of the multi-level cache memory system. Process 80 provides for invalidation of level one cache memory contents based upon conditions during the invalidate cycle. Process 82 is an enhancement based upon the validation cycle. Process 84 provides invalidation upon a parity error.

Process 80 is performed as a result of an instruction processor write request (i.e., an operand write) or memory bus SNOOP (i.e., monitoring of memory busses involving other instruction processors and other system controllers). For either of these activities, the corresponding data within the level one operand cache memory is invalidated under the shown circumstances. This invalidation during the invalidation cycle provides a time saving over a subsequent potential future request for the invalidated data.

Element 86 determines whether a write request has been made of the system controller (see also Fig. 3). Again, this request means that the instruction processor has attempted an operand write. If yes, element 88 determines whether this is a mode 3. If yes, element 92 requires ownership of the data within the level one operand cache memory or control is returned to element 94 without action. Note that a mode 3 write without ownership means that

there is no data within the level one operand cache memory to invalidate. Therefore, further activity within process 80 would be wasted.

However, if element 92 determines that ownership is present, control is given to element 90 which determines whether there is a level one cache memory hit (i.e., operand write is to a data element present within the level one cache memory). If there is a hit, element 102 invalidates the contents of the corresponding location with the level one operand cache memory. However, if there is no hit, there can be no time saving, because a memory operation involving the level three cache memory is required.

The other condition for which process 80 is applicable is a system memory bus SNOOP. In this activity, the system controller monitors the system memory buses for activity from other processors and system controllers which might impact the validity of its local data. The above identified commonly assigned, co-pending, and incorporated patent applications discuss the SNOOP activity in greater detail.

The SNOOP activity is sensed by element 94. When present, after checking for Mode 3, control is given to element 96 to determine whether it is mode 1. Element 100 determines whether the SNOOP experiences a hit on the bus read line (BRL). If no, control is returned to element 104, because modes 2 and 3 are not concerned with the bus read line. Element 98 determines whether there is a hit on the bus write line (BWL) or bus read invalidate line (BRIL).

If there is a hit, element 102 invalidates the corresponding location within the level one cache memory. This improves efficiency, because the data is invalidated without even being requested.

Read access is the concern of process 82. Element 104 determines when a read occurs. This is a result of an operand read operation within the instruction processor which experiences a level one operand cache memory miss. The request is made to the system controller wherein element 104 gives control to element 106. If there is also a miss within the level two cache memory, control is given to element 108 for retrieval of the requested data from the level three cache memory. After receiving the requested data, element 108 also loads the requested data into the level two cache memory and records its presence and location within the level one cache memory, thus making it potentially available for in response to a subsequent request.

The purpose of process 84 is an error recovery technique. Whenever element 112 determines that a parity error has been experienced with regard to the level two cache memory. When a parity error occurs, element 110 causes invalidation of the corresponding locations within the level one cache memory, to avoid loss of control between the instruction processor level one cache memories and the system controller level two cache memory.

Having thus described the preferred embodiments in sufficient detail for those of skill in the art to make and use the present invention, those of skill in the art will be readily able to apply the teachings found herein to yet other embodiments within the scope of the claims hereto attached.

**WE CLAIM:**

1. In a data processing system having a system bus and having a processor with a level one cache memory responsively coupled to a level two cache memory which is responsively coupled to a level three cache memory and having a circuit for SNOOPing said system bus , the improvement comprising:

a. First logic which invalidates a corresponding level one cache memory location in response to either a non-local memory write or write ownership request.

2. A data processing system according to claim 1 further comprising second logic which inhibits said first logic from invalidating for mode 3 requests without ownership.

3. A data processing system according to claim 1 further comprising:

a. Third logic which invalidates said corresponding cache memory location in response to a SNOOP hit.

4. A data processing system according to claim 3 further comprising:

a. Fourth logic which retrieves and records data in response to a level one cache memory read miss and a level two cache memory read miss.

5. A data processing system according to claim 1 further comprising:

a. Fifth logic which determines when said level two cache memory generates a parity error and which in response invalidates said corresponding level one cache memory location.

6. A data processing system comprising:

a. A processor having a level one cache memory;

b. A level two cache memory responsively coupled to said level one cache memory;

c. A level three memory responsively coupled to said level two cache memory; and

d. A first circuit which invalidates a corresponding portion of said level one cache memory in response to a level one cache memory write hit and a level two cache memory write.

7. A data processing system according to claim 6 further comprising:

a. A second circuit which inhibits said first circuit from said invalidating in response to a mode 3 lack of ownership.

8. A data processing system according to claim 6 further comprising:

a. A system memory bus;

b. A third circuit which SNOOPs said system memory bus; and

c. A fourth circuit which invalidates said corresponding portion of said level one cache memory in response to a SNOOP hit.

9. A data processing system according to claim 6 further comprising:

a. A fifth circuit which retrieves and records data in response to a level one cache memory read miss and a level two cache memory read miss.

10. A data processing system according to claim 6 further comprising:

a. A sixth circuit which detects parity errors of said level two cache memory and invalidates said corresponding portion of said level one cache memory in response to said detected parity error.

11. A method of maintaining validity of data within a level one cache memory of a processor responsively coupled to a level two cache memory which is responsively coupled to a system memory bus comprising:

a. Formulating a write memory request;

b. First experiencing a level one cache memory miss in response to said write memory request;

c. Second experiencing a level two cache memory hit in response to said first experiencing step; and

d. Invalidating a portion of said level one cache memory

corresponding to said write memory request in response to said second experiencing step.

12. A method according to claim 11 further comprising:

a. Inhibiting said invalidating step if said write memory request is mode 3 lacking ownership.

13. A method according to claim 11 further comprising:

a. SNOOPing said system memory bus; and

b. Invalidating said portion of said level one cache memory if said SNOOPing step identifies data corresponding to said write memory request.

14. A method according to claim 11 further comprising:

a. Formulating a read memory request;

b. Third experiencing a level one cache memory read miss; and

c. Retrieving and recording data corresponding to said read memory request.

15. A method according to claim 11 further comprising:

a. Determining whether a reference to said level two cache memory has a parity error; and

b. Invalidate said portion of said level one cache memory in response to said determining said parity error.

16. An apparatus comprising:

a. Means for executing program instructions;

b. Means responsively coupled to said executing means for level one caching data;

c. Means responsively coupled to said executing means and said level one caching means for accessing a data element if said executing means requires accessing of said data element and said level one caching means does not contain said data element;

d. Means responsively coupled to said requesting means for level two caching data; and

e. Means responsively coupled to said level one caching means for invalidating said data element if said data element is a write data element located within said level two caching means.

17. An apparatus according to claim 16 further comprising:

a. Means responsively coupled to said invalidating means for inhibiting said invalidating if said data element is mode 3 without ownership.

18. An apparatus according to claim 16 further comprising:

a. Means responsively coupled to said level two caching means for bussing system memory data;

b. Means responsively coupled to said bussing means for SNOOPing said bussing means; and

c. Means responsively coupled to said SNOOPing means for

invalidating said data element if said SNOOPing means locates a corresponding data element and said data element is a write data element.

19. An apparatus according to claim 16 further comprising:

a. Means responsively coupled to said level two caching means for retrieving and record said data element if said data element is a read data element and said level two caching means experiences a miss.

20. An apparatus according to claim 16 further comprising:

a. Means responsively coupled to said level two caching means for detecting a parity error; and

b. Means responsively coupled to said level one caching means and said detecting means for invalidating said data element if said detecting means detects said parity error.

# METHOD FOR IMPROVED FIRST LEVEL CACHE COHERENCY

## ABSTRACT OF THE DISCLOSURE

A method of and apparatus for improving the efficiency of a data processing system employing a multiple level cache memory system. The efficiencies result from invalidating level one cache information based upon a level one cache memory write. Similarly, the invalidation can occur from system bus SNOOPs. In addition, level one and level two cache memory misses result in loading and recording of the requested into both level one and level two cache memories. Furthermore, a level two cache memory parity error results in invalidation of the corresponding level one cache memory data.

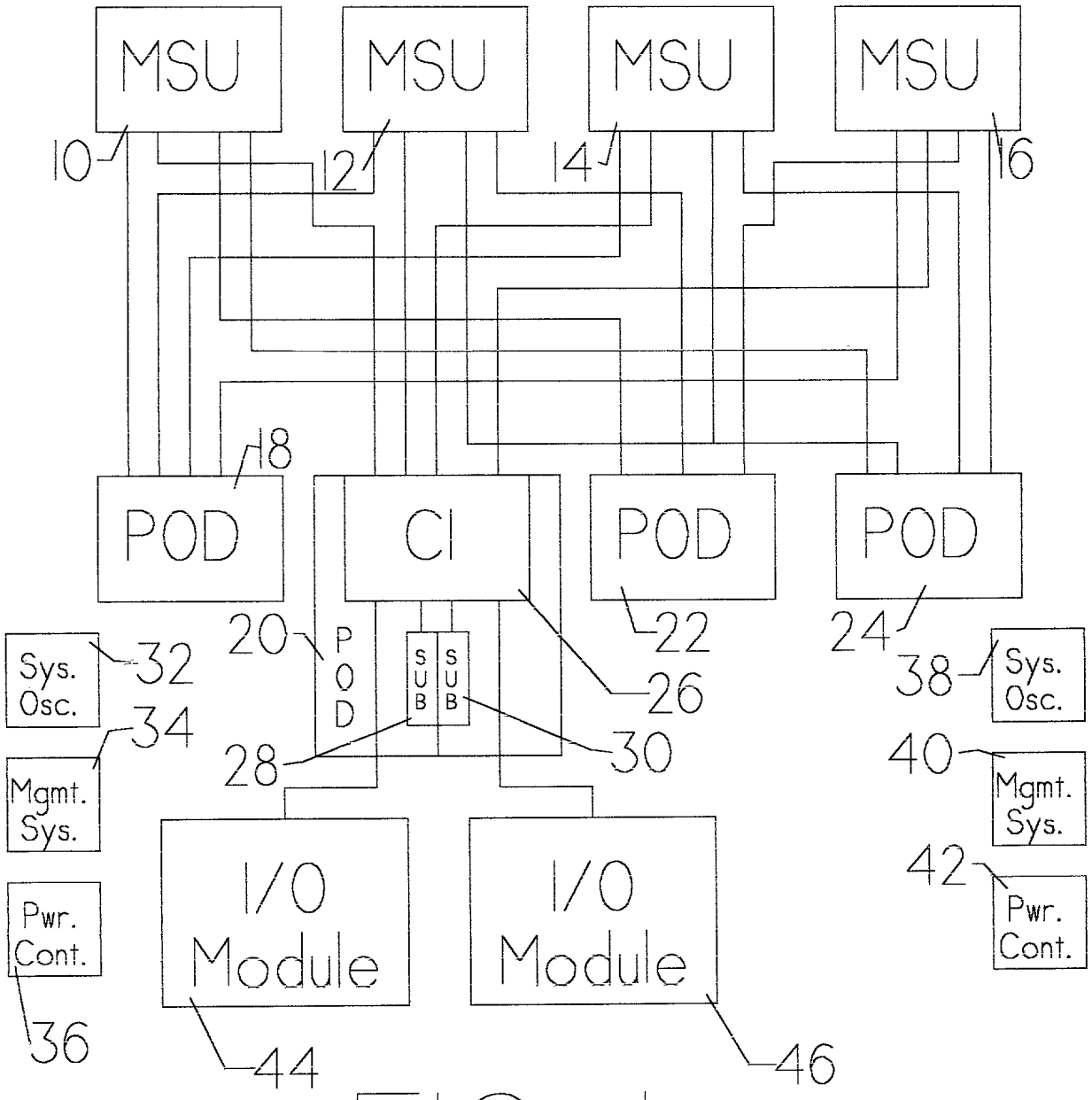

MSU
MSU
MSU
MSU
10
12
14
16
18
POD
CI
POD
POD
20
POD
SUB
SUB
22
24
26
28
30
Sys. Osc.
32
34
Mgmt. Sys.
Pwr. Cont.
36
I/O Module
I/O Module
44
46
38
Sys. Osc.
40
Mgmt. Sys.
42
Pwr. Cont.

FIG. 1

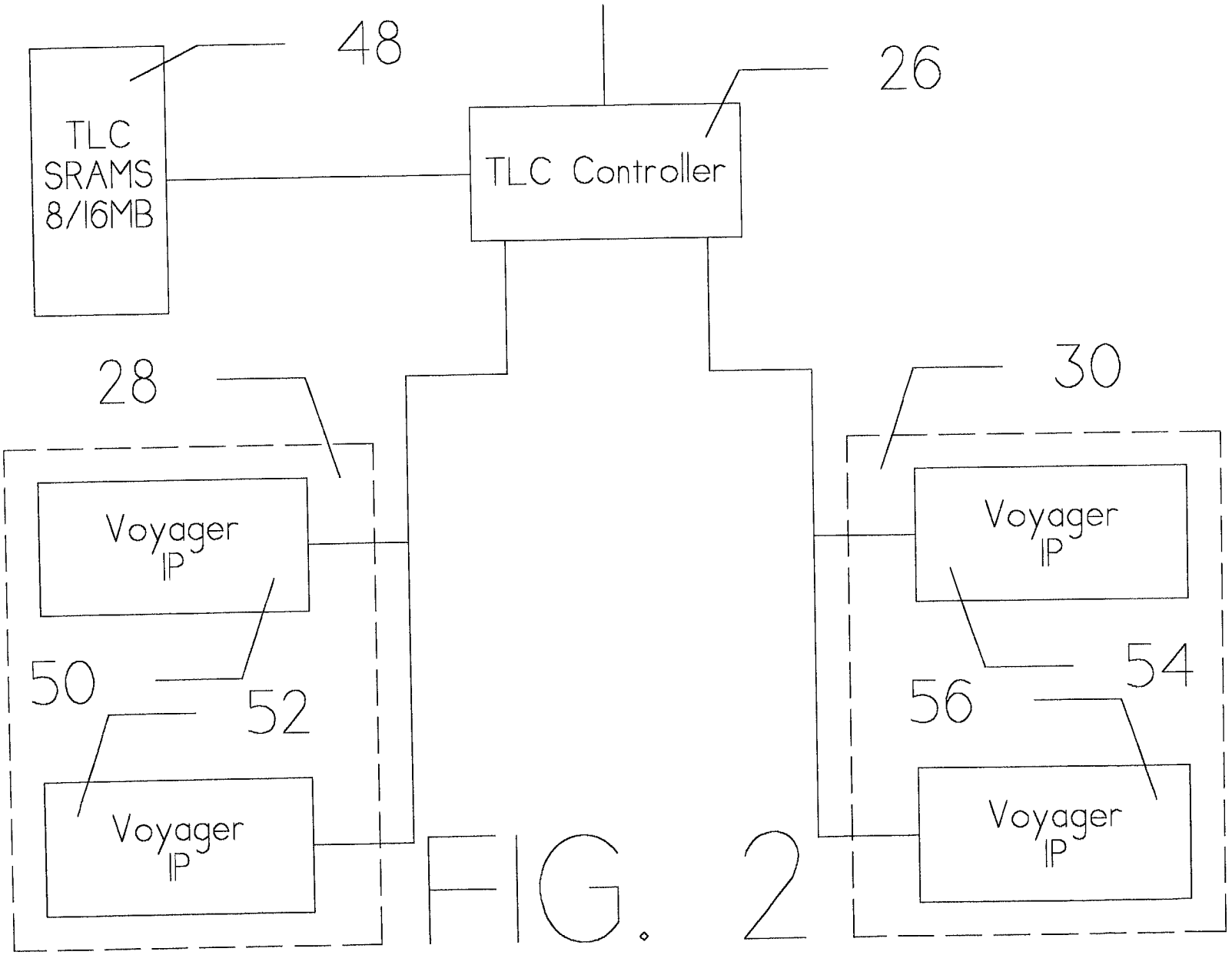

48
TLC
SRAMS
8/16MB
26
TLC Controller
28
30
Voyager
IP
Voyager
IP
50
52
54
56
Voyager
IP
Voyager
IP

FIG. 2

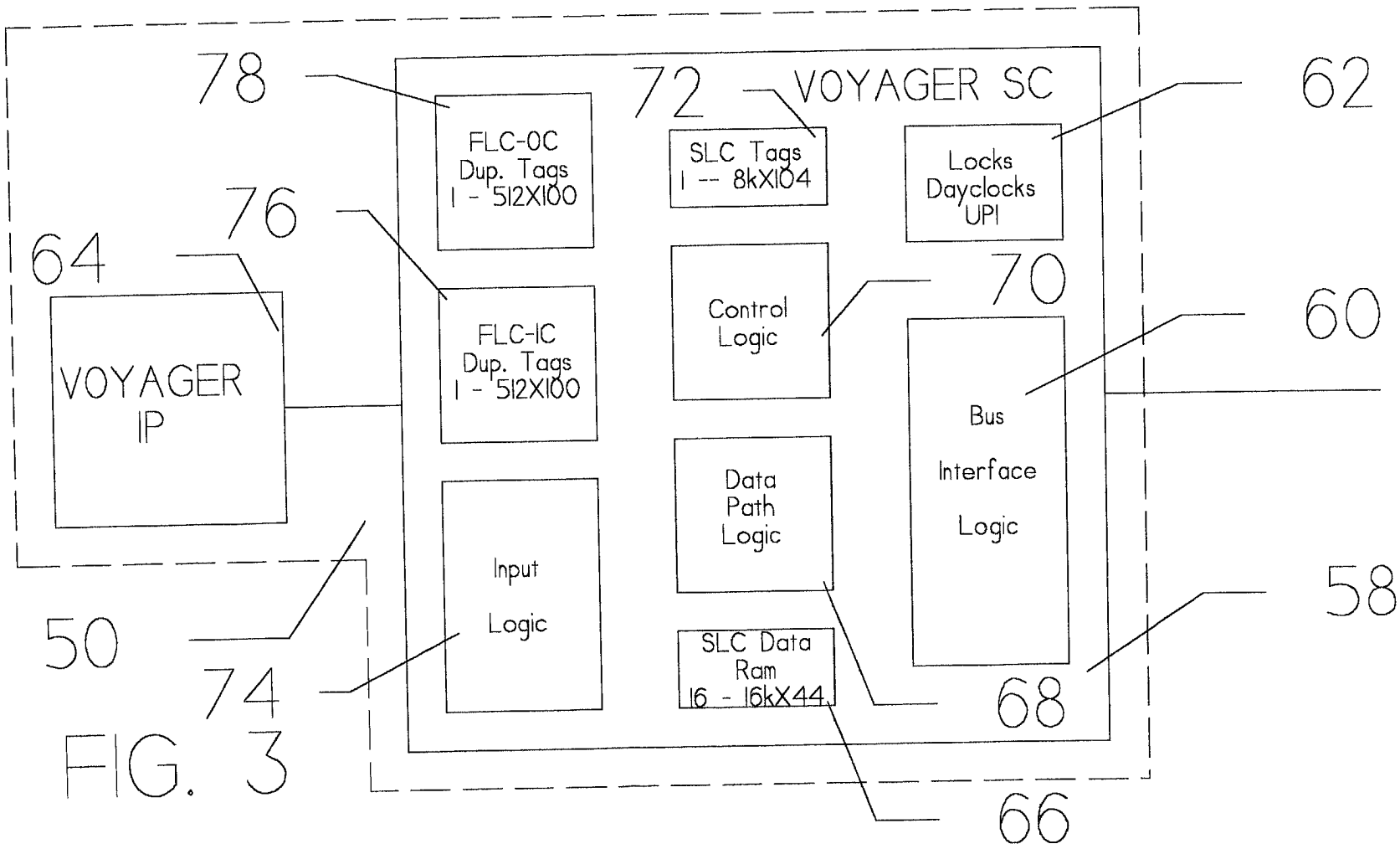

78
72
VOYAGER SC
62
FLC-0C
Dup. Tags
1 - 512X100
SLC Tags
1 -- 8kX104
Locks
Dayclocks
UPI
76
64
70
60
Control
Logic
VOYAGER
IP
FLC-IC
Dup. Tags
1 - 512X100
Bus
Interface
Logic
Data
Path
Logic
58
Input
Logic
50
74
SLC Data
Ram
16 - 16kX44
68
66

FIG. 3

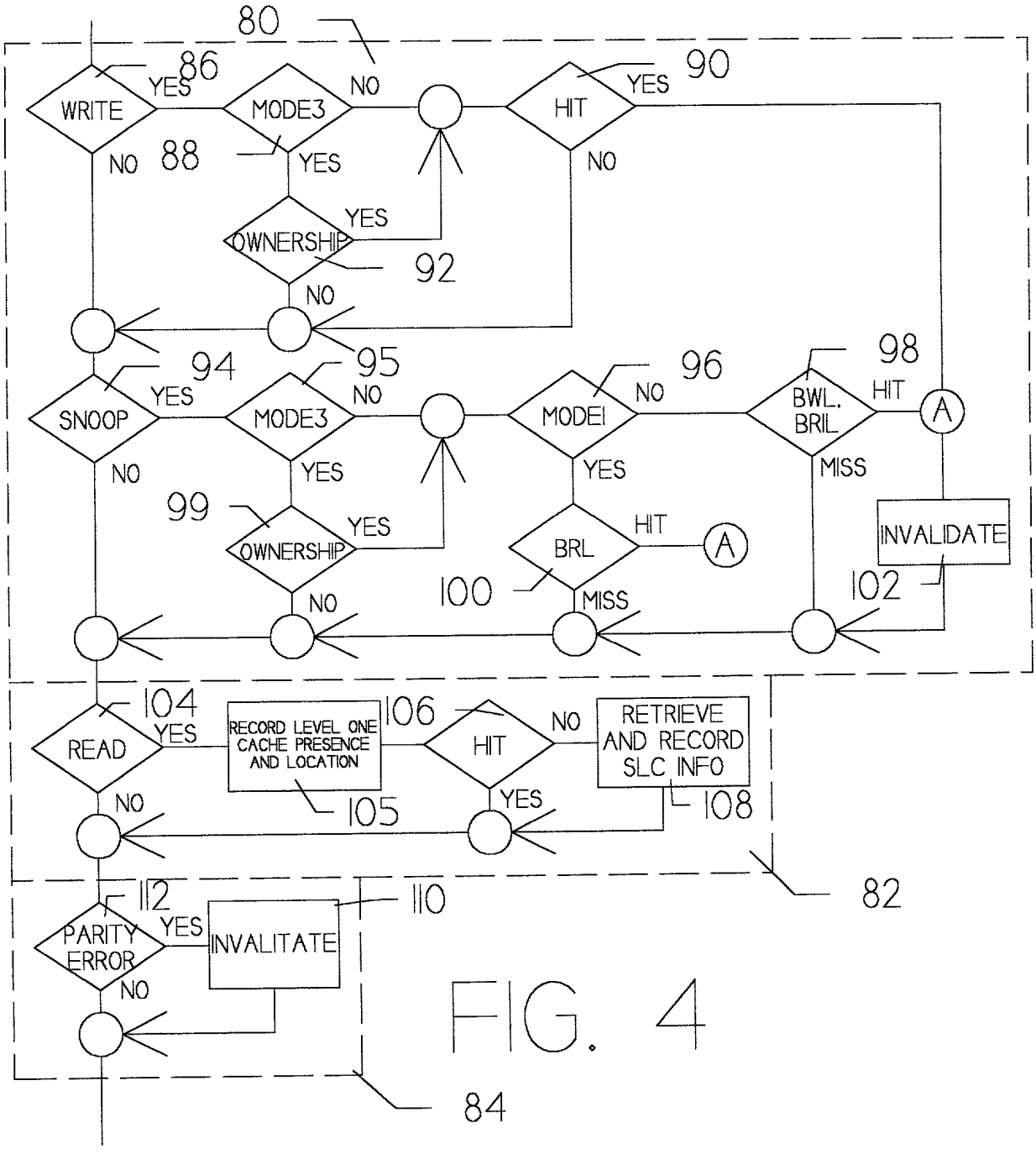

80
WRITE
86
YES
NO
MODE3
NO
YES
88
OWNERSHIP
YES
92
NO
HIT
90
YES
NO
SNOOP
94
YES
NO
MODE3
95
NO
YES
99
OWNERSHIP
YES
NO
MODEI
96
NO
YES
BRL
HIT
A
100
MISS
BWL.
BRIL
98
HIT
A
MISS
INVALIDATE
102
READ
104
YES
NO
RECORD LEVEL ONE CACHE PRESENCE AND LOCATION
105
HIT
106
NO
YES
RETRIEVE AND RECORD SLC INFO
108
82
PARITY ERROR
112
YES
NO
INVALITATE
110
84

FIG. 4

**COMBINED DECLARATION/POWER OF ATTORNEY FOR PATENT APPLICATION**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled METHOD FOR IMPROVED FIRST LEVEL CACHE COHERENCY, the specification of which (check one)

XX is attached hereto

__ was filed on _____
as U.S. Application
Serial No. _____

__ and was amended on (if
applicable) _____

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

I hereby claim foreign priority benefit(s) under Title 35, United States Code §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

| Prior Foreign Application(s) | | | Priority Claimed | |
|---|---|---|---|---|
| (Number) | (Country) | (Day/Month/Year Filed) | YES | NO |
| (Number) | (Country) | (Day/Month/Year Filed) | YES | NO |
| (Number) | (Country) | (Day/Month/Year Filed) | YES | NO |

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner

provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

| (Serial No.) | (Filing Date) | (Status) (patented, pending, abandoned) |
|---|---|---|
| | | |
| (Serial No.) | (Filing Date) | (Status-patented, pending, abandoned) |

**POWER OF ATTORNEY:** As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John L. Rooney, Reg. No. 28,898;
Lawrence M. Nawrocki, Reg. No. 29,333;
Wayne A. Sivertson, Reg. No. 25,645;
Richard C. Stempkovski, Jr., Reg. No. 45,130;
Jeffery L. Cameron, Reg. No. 43,527;
Donald A. Jacobson, Reg. No. 22,308; and
Charles A. Johnson, Reg. No. 20,852

Send correspondence to:

Charles A. Johnson
Unisys Corporation
Law Department
M.S. 4773
2470 Highcrest Road
Roseville, Minnesota 55113

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon, I further declare that I understand the content of this declaration.

Full name of sole or first inventor Paul S. Neuman
Inventor's Signature Paul Neuman Date 8/15/2000
Residence 1542 Sherwood Road
Shoreview, Minnesota 55126 Citizenship U.S.A.
Post Office Address 1542 Sherwood Road
Shoreview, Minnesota 55126

**1.56 Duty to disclose information material to patentability.**

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is cancelled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is cancelled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) the closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made of record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or
(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application:

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.